IN THE CLAIMS

1. (canceled)
2. (canceled)
3. (canceled)
4. (canceled)
5. (canceled)
6. (canceled)

7. (previously presented) The simulation environment of claim 41, wherein the control process sets up a predetermined ordered queue of the server processes and a predetermined ordered queue of the client processes, and the messages are sent to and from the client processes and the server processes according to the predetermined ordered queues of server and client processes.

8. (previously presented) The simulation environment of claim 41, wherein:

    the server process evaluates an event expression to determine the occurrence of an event in the server process, and,

    the server process sends an event expression message to the control process upon the occurrence of the event in the server process, the event expression message containing a time stamp, the time stamp being an indication of a time at which the event occurred in the server process.

9. (canceled)

10. (previously presented) The simulation environment of claim 41, wherein the control process maintains a time stamp for each server process, the time stamp being an indication of an elapsed time from a start of the control process, where the elapsed time is proportional to a time elapsed in the control process between the synchronization points.

11. (previously presented) The simulation environment of claim 41, wherein the control process sets up a server order queue of the server processes and a client order queue of the client processes, and the messages are sent to and from the

client processes and the server processes according to the server order queue and the client order queue.

12. (previously presented) The simulation environment of claim 11, wherein:

the control process receives a plurality of the event expression messages from the server processes,

the control process sorts the event expression messages received from the server processes according to the server order queue, and

the control process orders each of the event expression messages within the server order queue according to an earliest of the time stamps at which the event occurred in the server process.

13. (previously presented) The simulation environment of claim 12, wherein the control process delivers the sorted event expression messages to the client processes associated with the server processes according to the client order queue.

14. (previously presented) The simulation environment of claim 41, wherein:

the client processes each send a finish message, indicating the client process is finished running, to the control process for communication to the server process associated with the client process, when the client process is finished running,

the control process holds each of the finish messages from the client processes until all of the client processes associated with a server process are finished running, and

the control process sends a finish message to the server process indicating the client processes are finished running.

15. (previously presented) The simulation environment of claim 41, wherein:

the server processes each send a finish message, indicating the server process is finished running, to the control process when the client processes associated with each of the server processes are finished,

5   the control process holds each of the finish messages from the server processes until all of the server processes have sent the finish messages to the control process, and

the server processes, client processes, and control process finish operations and exit.

16.   (previously presented) The simulation environment of claim 41, further comprising:

a plurality of client processes, each of the client processes associated with a predetermined server process, and communicating with the predetermined server process under the direction of the control process, and

5   a plurality of server processes, each of the server processes evaluating an event expression to determine the occurrence of an event in the server process, and each of the server processes sending an event expression message to the control process upon the occurrence of the event in the server process,

10   the event expression message containing a time stamp indicating a time at which the event occurred in the server process.

17.   (previously presented) The simulation environment of claim 41, further comprising the control process setting up a client ordered queue of client processes, a server ordered queue of server processes, and a time ordered queue of event expression messages received from the server processes, the time ordered

5   queue ordered according to an earliest in time event expression message.

18.   (canceled)
19.   (canceled)

20.   (previously presented) The simulation environment of claim 41, wherein the simulation environment simulates a device selected from a group consisting of electrical devices, mechanical devices, electromechanical devices, computer networks, DSL modems, ASIC disk drive controllers, graphic processors, network

5   interface adapters, and communications networks.

21.   (canceled)

22. (previously presented) The simulation environment of claim 41, wherein the control process includes a synchronization varying module for varying an elapsed time duration between the synchronization points.

23. (previously presented) The simulation environment of claim 41, wherein the control process stops the server process when the server process reaches a synchronization point.

24. (canceled)
25. (canceled)
26. (canceled)

27. (previously presented) The simulation environment of claim 41, wherein:

a plurality of server processes, a plurality of client processes associated with the server processes, the plurality of server processes communicating via the control process with the client processes associated with each of the server processes,

wherein each of the server processes evaluates an event expression to determine the occurrence of an event in the server process, and each of the server processes sends an event expression message to the control process upon the occurrence of the event in the server process, the event expression message contains a time stamp indicating a time at which the event occurred in the server process.

28. (canceled)
29. (canceled)

30. (withdrawn) The method of claim 42, further comprising the steps of:

determining the occurrence of predetermined events in the server processes,

maintaining with the control process a list of client processes, a list of server processes, and a list of messages associated with the predetermined events, and

communicating the associated message to the client processes upon occurrence of one of the predetermined events.

31.   (withdrawn) The method of claim 30, further comprising the steps of:

ordering with the control process the messages according to an earliest time that the predetermined events occurred in the server processes, and

delivering the messages to the client processes according to the ordering.

32.   (withdrawn) The method of claim 31, wherein the ordering of the messages is determined by at least one of:

(1) time order, by an earliest time that such predetermined events occurred in the server processes,

(2) server order, according to a predetermined order of server processes, and

(3) client order, according to a predetermined order of clients.

33.   (withdrawn) The method of claim 32, further comprising the steps of:

sorting the messages according to the server order and the time order, and

delivering the messages from the control process to the client processes according to the client order and the time order, with earliest messages delivered first.

34.   (canceled)
35.   (canceled)

36.   (withdrawn) The method of claim 42, further comprising the steps of:

setting a plurality of synchronization points of elapsed time in the simulation of servers and clients,

determining an occurrence of a predetermined event in the server processes,

maintaining, with the control process, a list of client processes, a list of server processes, and a list of occurrences of the predetermined events,

communicating the predetermined events to the client processes,

ordering with the control process the predetermined events according to an earliest time that the predetermined events occurred in the server processes, and

delivering messages to the client processes relating to the predetermined events according to the ordering of the predetermined events.

37.   (canceled)

38.   (withdrawn) The method of claim 42, further comprising the steps of:

polling each of the client processes with the control process in a predetermined manner,

temporarily storing the messages from the client processes until the client processes issue a predetermined message to simulate to the control process, and

forwarding the messages from the client processes to the server processes associated with the client processes upon the occurrence of the predetermined message to simulate.

39.   (canceled)
40.   (canceled)

41.   (previously presented) A simulation environment running on a computer system comprising:

at least one server process capable of sending and receiving messages,

at least one client process capable of sending and receiving messages, and

only one control process for receiving the messages sent from the server process and the client process and for sending the messages sent from the server process and the client process, where all messages between the server process and the client process are controlled by and relayed through the control process,

the control process sets synchronization points in the server process, the synchronization points indicating points in time where the server process pauses until restarted by the control process,

where the server process, the client process, and the control process are all separate and distinct processes.

42.   (withdrawn) A method for simulating a process in a simulation environment running on a computer system, the method comprising the sequential steps of:

a.   starting only one control process within the simulation environment,

b.   providing configuration information to the control process, the configuration information including a number and function of server processes to include in the simulated process, and a number and function of client processes to include in the simulated process, where the number of server processes is at least one and the number of client processes is at least one, and the number of server processes and the number of client processes are mutually independent,

c.   starting the server processes included in the simulated process with the control program,

d.   setting a synchronization point in the server processes with the control program,

e.   sending a ready to synchronize message from each server process to the control program, indicating that the server process sending the ready to synchronize message has started,

f.   continuing the method when each server process included in the simulated process has sent a ready to synchronize message to the control program,

g.   starting the client processes included in the simulated process with the control program,

h.   polling each client process for simulate messages with the control program, using a predetermined polling order that does not vary for a given simulation,

i.   accepting the simulate messages from the client processes with the control program in response to the polling of the client processes,

j.   forwarding the simulate messages from the control program to an appropriate one of the server processes,

k.   when simulate messages have been received from all of the client processes included in the simulated process, processing the simulate messages with the server processes,

l.   sending synchronization point reports to the control process with the server processes, indicating that each server process sending a synchronization point report has reached its synchronization point, where

35   further processing of server processes reaching their synchronization point is paused, and

m.   when the control program has received synchronization point reports from all of the server processes included in the simulated process, restarting the method at step (h), until the simulated process is completed.

43.   (withdrawn) The method of claim 42, further comprising, in step (m), setting a different common synchronization point in all the server processes before restarting the method.

44.   (withdrawn) The method of claim 42, wherein a common synchronization point is sent to each server process.

45.   (withdrawn) The method of claim 42, wherein the method further comprises between step (k) and step (l):

receiving a user specified event with at least one of the server processes,

creating an event message with a time stamp with each of the at least one server
5   processes,

sending the event message to the control program with each of the at least one server processes,

pausing further processing of each of the at least one server processes,

holding the event messages with the control process until all server processes
10   have sent one of an event message and a synchronization point report,

when the control process has received one of an event message and a synchronization point report from each server process, then sending the event messages to the client processes,

acting on the event messages with the client processes, and
15   when all event messages have been acted on by the client processes, then sending a simulate message to server processes that have not yet sent synchronization point reports, which simulate messages instruct the server processes that have not yet sent synchronization point reports to continue processing.

46.   (withdrawn) The method of claim 42, wherein the simulation process is completed by:

sending finish messages to the control program with the client processes,

sending a finish message to one of the server processes when all of the client processes associated with the one of the server processes has sent finish messages,

responding to the finish message sent to the one of the server processes with the one of the server processes,

when all of the server processes have received a finish message from the control program and all of the server processes have responded to the finish message, then

exiting the server processes,

exiting the client processes, and

exiting the control process.